MVC 4.0, Part 10 of 11: Dependency Injection and Deployment

page 1

Meet the expert: Philip Ledgerwood has been a software developer for fifteen years. He currently works primarily in .NET technologies producing custom software for organizations of all sizes. He has also done extensive training for those same organizations in both technical and business process topics. Philip is a strong advocate of Lean and agile software development and spends most of his time helping companies interested in the value those practices can bring to their development efforts. He does this through a combination of training and working "in the trenches" as a developer on these teams, keeping a hand in the academic side of emerging technology and practices while also directly applying it in real projects to bring real business value.

Prerequisites: This course assumes that you are familiar and experienced with Microsoft's .NET Framework and ASP.NET development tools. You should be familiar with Web development and understand how HTTP and HTML work to produce Web pages for the user. You should have experience writing applications with ASP.NET 4.0 or later Web forms, and be familiar with how ASP.NET processes page requests, and have strong experience with .NET Framework 4.0 or later programming. You should have experience with Visual Studio 2012 for building Web application projects. Experience with building database applications using these tools will be helpful, although not strictly necessary.

Runtime: 02:01:44

Course description: In this course we'll talk about what dependence injection is and what kinds of problems it's trying to solve. We'll take a look at what every dependence injector needs to do when you wire it up to your MVC framework application, including how you might go about writing your own dependence resolver. Next we are going to take a look at integration testing where we test the entire web application in an automated way, starting from the user interface and drilling down through all the functionality. We are going to take a look at unit testing and where we can isolate down to the very controller methods and make sure those are working properly. Finally we'll take a look at where we can put these applications and how easy it is with MVC applications to deploy them, ranging all the way from simply coping your files to the very elaborate and extensive and handy functionality that is built in to the publish functionality built in Visual Studio.

Course outline:

Dependency Injection
- Introduction
- Dependency Injection
- What Is It
- Demo: Logging
- Demo: Logger outside
- Summary

Resolvers
- Introduction
- Resolving Dependencies
- Demo:Structure Map
- Demo: Wired to Framework
- Summary

DI Frameworks
- Introduction
- Common DI Frameworks
- Using DI Frameworks
- Demo: DI Framework
- Demo: Structure Map
- Demo: SQL Server Logger
- Demo: MVC Applications
- Demo: Unit Tests
- Demo: MSpec
- Summary

Integration Testing
- Introduction
- Integration Testing
- Demo: Integration Testing
- Demo: Test tools
- Demo: Write Test
- Demo: Feature Tested
- Demo: Make Steps
- Demo: How Does it Work
- Demo: Run Test
- Summary

Unit Testing
- Introduction
- Unit Testing
- Demo: Unit Testing
- Demo: Adding Unit Test
- Demo: Home Controller
- Demo: Test Patterns
- Demo: Check
- Summary

Exception Handling
- Introduction
- Exception Handling
- Demo: Exception Handling
- Demo: Error Status Code
- Summary

Deployment
- Introduction
- Deployment
- Demo: VS Deployment
- Demo: FTP Deployment
- Demo: Hosting Gallery
- Demo: FPSE
- Demo: File System
- Summary